

## REMARKS

In the final Office Action, the Examiner continues to reject the claims under 35 USC § 251 and under 35 USC §102. These objections and rejections are fully traversed below. Applicant notes that the rejection of claims 1-73 under 35 USC §102(e) as anticipated by Adams Jr. et al. and the rejection of claims 1, 11, 18, 20, 22, 44, 54, 61, and 65 under 35 USC § 112 are withdrawn. In addition, the Examiner has indicated that claims 1-31, 34-39, 54-59, and 69-73 are allowed.

Claims 1-73 remain pending. Reconsideration of the application is respectfully requested based on the following remarks.

### REJECTION OF CLAIMS 40-53 AND 60-68 UNDER 35 USC §251

In the Office Action, the Examiner rejected claims 40-53 and 60-68 under 35 USC §251 as being an improper recapture of claimed subject matter cancelled in the application for the patent upon which the present reissue is based. As pointed out by the Examiner, original claims 6 and 14 were amended to specifically require that a new header be generated when the data packet is encrypted. As is understood by the Examiner, original claims 6 and 14 were directed at encryption of the data packet while the rejected claims 40-53 and 60-68 are directed to decryption rather than encryption. The Examiner has taken the position that since decryption is the complementary process to encryption, "the decryption process claimed must contain the second header as well" to avoid recapturing the material implicitly. Therefore, the Examiner has implied that the encryption process requires the generation of a second header.

In prior communications, Applicant has argued that claims to decryption need not recite the generation of a new address header. In the final office action dated January 5, 2000, the Examiner responded that although the step of "generating" is not appropriate, "the decryption process claimed must contain the second header as well" to avoid recapturing the

material implicitly. In other words, the Examiner argues that the encryption step of the original claims 6 and 14, as amended, requires a “second header.” However, the encryption step of original claims 6 and 14 does not require two headers. More particularly, while original claim 14 recites an encapsulation header that includes the new address header, claim 6 recites only a “new address header.” Therefore, two different headers are not explicitly recited in the encryption step of original claims 6 and 14, as amended. It therefore appears that the Examiner is arguing that the use of the phrase “new address header” somehow implies that the encryption step requires two headers.

In no manner does the word “new” imply that the “**new** header” is a second header, as the Examiner has suggested. In construing claims, terms are generally given their plain meaning (*i.e.*, the ordinary and customary meaning given them by one skilled in the art) unless examination of the specification, prosecution history, and other claims indicate that the patentee intended otherwise. *Vitrionics*, 39 USPQ2d at 1576; *Nike Inc. v. Wolverine World Wide Inc.*, 33 USPQ2d 1038, 1039 (Fed. Cir. 1994); *Carroll Touch, Inc. v. Electro Mechanical Systems.*, 27 USPQ2d 1836, 1840 (Fed. Cir. 1993). In order to illustrate the ordinary and customary meaning of the term “new,” the patentee refers to Webster’s II New Riverside University Dictionary and Roget’s International Thesaurus, Fourth Edition. Roget’s International Thesaurus, Fourth Edition indicates that the word “new” is an adjective which can mean additional, fresh, original, and unused. Moreover, synonyms of each of these meanings are explicitly set forth. For example, synonyms of the term “additional” include supplemental, extra, other, and another. As another example, synonyms of the term “fresh” include untouched and unhandled. Similarly, Webster’s II New Riverside University Dictionary, indicates that the word “new” has the following meanings: 1. “Having existed or been made for only a short time” 2. a. “Not yet old” 3. “Just discovered, found, or learned” 4. “Unfamiliar: novel” 5. “Starting again in a cycle” ...and 9. “Changed for the better.” There is no indication in the specification or prosecution history that the patentee intended otherwise.

The prosecution history does not support the Examiner’s assertions. In the communication dated January 11, 2001, the Examiner refers to claims 16 and 17, which were newly presented as part of the amendment dated December 21, 1995. As indicated by the Examiner, claims 16 and 17 specifically recite a “new address header.” Again, the arguments set forth above with respect to the phrase “new address header” apply here. Similarly, the Examiner refers to “Applicant’s arguments” on page 5 of the amendment dated

December 21, 1995, which the Examiner states “rely on the new header for overcoming the art rejection.” The “Applicant’s arguments” set forth on page 5 of the amendment do refer to a “new address header” that is appended to a data packet. However, in no manner does the Applicant argue or imply that the “new address header” is a second address header or require that an encapsulation header that includes the new address header be appended to the data packet. Moreover, there is no indication in the prosecution history that the patentee intended that the phrase “new header” mean “second header.” The Examiner appears to have limited the meaning of the term “new” without any basis for such a limitation in the prosecution history. Although it is possible that the new header may be a second header that is appended to the data packet as the Examiner suggested, the new header may also merely be a replacement header to an already existing header to the data packet. For example, referring to the definitions set forth above, the new header may simply be a header that is “changed for the better.” The Examiner argues that the appending of such a replacement header is “a situation in which recapture would be avoided but...does not exclude recapture based on previous response.” Applicant respectfully traverses this assertion. The instance in which the new header is a replacement header is not only one “situation” in which recapture would be avoided. Rather, this example is merely an illustration of one potential meaning of the phrase “new header” in view of the prosecution history, and serves to illustrate that the phrase “new header” does not require the generation of a “second header.” The only requirement imposed by the amendments to original claims 6 and 14 is that a “new header” be generated and appended to the data packet. Thus, this “new header” need not be a “second header.” On the contrary, the “new header” may be the sole header of the data packet resulting from application of the method of claims 6 and 14. Accordingly, the generation of a “new address header” does not require or imply that a **second header** is generated during **encryption**.

The first step in applying the recapture rule is to determine whether and in what aspect the reissue claims are broader than the patent claims. As discussed above, the encryption step does require that a **new header** be generated. However, the encryption step does not require that a **second header** (or two headers) be generated. Therefore, the fact that the decryption step does not recite a second header does not broaden the reissue claims in this respect. It follows that application of the second step in which it is determined whether the broader aspects of the reissue claims relate to surrendered subject matter becomes moot. Accordingly, the absence of the recitation of a second header in the decryption step does not

“recapture” claimed subject matter surrendered in the application to obtain the original patent.

Specifically, claims 40-53 and 60-68 do recite a header. More particularly, as described above, each of claims 40-53 and 60-68 recite receiving a data packet including a header section and a data section. The feature they do not recite is where the header is generated. It is acknowledged that in many (and probably most) situations, the header that is on the decrypted packet will have been generated by the encryption source. However, this is by no means a requirement. From the standpoint of the device handling the decryption, it is typically irrelevant which process or mechanism appended the header to the data packet that is being decrypted. For instance, the fact that the header appended by the encryption process is a “new header” (e.g., changed for the better) from the viewpoint of the encryption process is irrelevant to the decryption process. Thus, the encryption and decryption steps are NOT necessarily mirror images of one another. Encryption and decryption steps are separate processes and, in this case, the use of the phrase “new header” is inappropriate with respect to claims directed solely to decryption. Of course, in the present case, since neither the encryption nor decryption steps require that the encrypted data packet include a second header, the encryption and decryption steps are mirror images of one another in this respect. Accordingly, it is respectfully submitted that amendments made to the encryption claims do not have an estoppel effect on the claims directed at decryption, and that the pending rejection of claims 40-53 and 60-68 under 35 USC §251 should be reversed.

#### **REJECTION OF CLAIMS 32-33, 40-41, AND 44-53 UNDER 35 USC §102**

Claims 32-33, 40-41, and 44-53 stand rejected under 35 USC §102(b) as being anticipated by White. In the Office Actions, the Examiner has argued that the Site Address (for entry into the WAN) disclosed in White is a network address which is the same as the “broadcast address” as defined in Applicant’s specification. Applicant respectfully traverses this assertion. Col. 4, lines 10-15 of White state that the header contains Site Address information that identifies a node via which the packet enters and leaves the WAN. Thus, the Site Address disclosed in White refers to a specific node rather than a “broadcast address,” as will be described in further detail below. In the communication dated January 11, 2001, the Examiner admits that a “[b]roadcast address refers to all host [sic] on the

network.” It is also important to note that the term “broadcast address” has a very specific meaning to those of ordinary skill in the art, as will be described in further detail below. In no manner does White disclose or suggest that the site address is actually a “broadcast address.” As described above, column 4, lines 10-15 of White state that the header contains Site Address information that identifies a **node** via which the packet enters and leaves the WAN. In other words, White does not state that the header contains Site Address information that identifies all hosts on the network, as the Examiner suggests. Similarly, White does not state that the header contains Site Address information that includes a broadcast address of all hosts on the network. The identification of a **node** within the header as required by column 4, lines 10-15 of White does not anticipate claims that require the identification of a **broadcast address** in the header (or that merely identify a **network** in the header), which would refer to all hosts on the network rather than a specific node in the network.

#### Independent claims 40, 50, and 52

Each of independent claims 40, 50, and 52 pertain to a method, a computer program product, or a computer system for decryption of a data packet including a header section that stores a destination identifier of a broadcast address of the destination of the data packet and a source identifier of a broadcast address of the source of the data packet.

Applicant acknowledges that White discloses a header that identifies the node via which the packet enters and leaves the network (i.e., Site Address). See White, col. 4, lines 11-15. However, Applicant respectfully submits that the Site Address disclosed in White is not equivalent to a broadcast address. Claims 40, 50, and 52 each specifically require that the header section store a source identifier identifying a “broadcast address” of the source and a destination identifier identifying a “broadcast address” of the destination. The term “broadcast address” is known in the art to refer to an IP address used for transmitting packets to all hosts on a given network. In other words, through the use of a broadcast address, a single host cannot be identified. In practice, the host portion of a broadcast address typically contains all 1s or all 0s. White neither discloses nor suggests that the header identifies a broadcast address of the source and destination of the data packet. Rather, White requires that the header identify an actual node via which the packet enters and leaves the network, as

described above. The invention of claims 40, 50, and 52 prevents using the address of a particular node, particularly a node responsible for encryption or decryption of the data packet. The presently claimed invention therefore provides greater protection against tapping into the network to decipher the nature of the information transmitted. Accordingly, Applicant respectfully submits that White's disclosure of a header that identifies a node via which a packet enters and leaves a network does not anticipate claims requiring that a header section of a data packet include a source identifier identifying a "broadcast address" of the source and a destination identifier identifying a "broadcast address" of the destination.

Each of claims 40, 50, and 52 further require determining whether the data packet is encrypted upon reference to at least one of the source and destination identifiers and decrypting the data packet to produce a decrypted data packet if the data packet is encrypted. White neither discloses nor suggests determining whether a data packet is encrypted upon reference to at least one of the source and destination identifiers, which identify a "broadcast address" of the source and a "broadcast address" of the destination, respectively. Nor does White disclose or suggest decrypting the data packet upon such a determination. Therefore, it is respectively submitted that claims 40, 50, and 52 are not anticipated by White, and that the pending rejection of these claims should be reversed for the reasons set forth.

Dependent claims 41, 44-49, 51, and 53

Claim 41 depends directly from independent claim 40 and further comprises transmitting the decrypted data packet to the destination. Applicant respectfully submits that White neither discloses nor suggests transmitting a decrypted data packet to a destination which has been decrypted upon a determination made as described above with reference to claim 40. Accordingly, claim 40 is not anticipated by White and the pending rejection of claim 40 should be reversed for this reason as well.

Claim 44 depends directly from independent claim 40 and further requires that the data section of the data packet include an encrypted header section and an encrypted data section, the encrypted header section including a header of the decrypted data packet after encryption and the encrypted data section including a body of the decrypted data packet after encryption. White neither discloses nor suggests such an encrypted header section and encrypted data section. Thus, Applicant respectfully submits that claim 44 is patentable over

White, and respectfully requests that the rejection of claim 44 be reversed for this reason as well.

Claims 45-47 further depend from claim 44. Claim 45 depends from claim 44 and further recites wherein the encrypted header section stores the source and destination identifiers. White neither discloses nor suggests storing the source and destination identifiers (which identify broadcast addresses of the source and destination, respectively) in an encrypted header section. Thus, Applicant respectfully submits that claim 45 is not anticipated by White, and respectfully submits that the rejection of claim 45 should be reversed for this reason as well.

Claim 46 depends from claim 44 and further recites wherein the source is a network and the encrypted header section stores an identifier of a host computer in the network. White neither discloses nor suggests storing an identifier of a host computer in the network (which is the source of the packet) in an encrypted header. Thus, Applicant respectfully submits that claim 46 is not anticipated by White, and respectfully submits that the rejection of claim 46 should be reversed for this reason as well.

Claim 47 depends from claim 44 and further recites wherein the destination is a network and the encrypted header section stores an identifier of a host computer in the network. White neither discloses nor suggests storing an identifier of a host computer in the network (which is the destination of the packet) in an encrypted header. Thus, Applicant respectfully submits that claim 47 is not anticipated by White, and respectfully submits that the rejection of claim 47 should be reversed for this reason as well.

Claim 48 depends from claim 40 and further requires that the source is a host computer or a network. White neither discloses nor suggests performing the method of claim 40 where the source is a host computer or a network. Thus, Applicant respectfully submits that claim 48 is not anticipated by White, and respectfully submits that the rejection of claim 48 should be reversed for this reason as well.

Claim 49 depends from claim 40 and further requires that the destination is a host computer or a network. White neither discloses nor suggests performing the method of claim 40 where the destination is a host computer or a network. Thus, Applicant respectfully submits that claim 49 is not anticipated by White, and respectfully submits that the rejection of claim 49 should be reversed for this reason as well.

Claim 51 depends from claim 50 and further requires that the computer readable medium is a memory, random-access-memory, read-only-memory, disk drive, or CD-ROM.

White neither discloses nor suggests storing the computer program product of claim 50 in such a computer readable medium. Accordingly, Applicant respectfully submits that claim 51 is not anticipated by White, and respectfully submits that the rejection of claim 51 should be reversed for this reason as well.

#### Independent claims 32 and 33

Claim 32 pertains to a method of encryption that produces a modified data packet including a header portion that stores an identifier of the network of the source of the data packet. Similarly, claim 33 pertains to a method of encryption that produces a modified data packet including a header portion that stores an identifier of the network of the destination of the data packet. Thus, both claims 32 and 33 each specifically require that an identifier of a network be provided in a header portion of the data packet.

A single host cannot be identified through the mere identification of a network. As described above, White requires that the header identify an actual node via which the packet enters and leaves the network, as described above. In contrast, the invention of claims 32 and 33 prevents using the address of a particular node, particularly a node responsible for encryption or decryption of the data packet. The presently claimed invention therefore provides greater protection against tapping into the network to decipher the nature of the information transmitted. Accordingly, Applicant respectfully submits that White's disclosure of a header that identifies a node via which a packet enters and leaves a network does not anticipate claims requiring that a header section of a data packet include an identifier of a network of the destination or source of the data packet. Therefore, it is respectfully submitted that claims 32 and 33 are not anticipated by White, and that the pending rejection of these claims should be reversed for the reasons set forth.



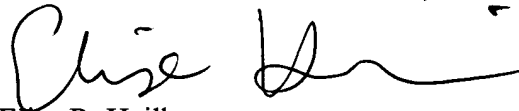
**SUMMARY**

Reconsideration of the application and an early Notice of Allowance are earnestly solicited. If there are any issues remaining which the Examiner believes could be resolved through either a Supplemental Response or an Examiner's Amendment, the Examiner is respectfully requested to contact the undersigned attorney at the telephone number listed below.

Applicants hereby petition for an extension of time which may be required to maintain the pendency of this case, and any required fee for such extension or any further fee required in connection with the filing of this Amendment is to be charged to Deposit Account No. 50-0388 (Order No. SUN1P342R).

Respectfully submitted,

BEYER WEAVER & THOMAS, LLP

A handwritten signature in black ink, appearing to read 'Elise', followed by a long, horizontal, wavy line that extends to the right.

Elise R. Heilbrunn

Reg. No. 42,649

BEYER WEAVER & THOMAS, LLP  
P.O. Box 778  
Berkeley, CA 94704-0778  
Tel: (510) 843-6200

## APPENDIX

1. (Once Amended) A method for transmitting and receiving packets of data via [a] an internetwork for a first host computer on a first computer network to a second host computer on a second computer network, the first and second computer networks including, respectively, first and second bridge computers, each of said first and second host computers and first and second bridge computers including a processor and a memory for storing instructions for execution by the processor, each of said first and second bridge computers further including memory for storing at least one predetermined encryption/decryption mechanism and information identifying a predetermined plurality of host computers as hosts requiring security for packets transmitted between them, the method being carried [carded] out [be] by means of the instructions stored on said respective memories and including the steps of:

- (1) generating, by the first host computer, a first data packet for transmission to the second host computer, a portion of the first data packet including information representing an internetwork address of the first host computer and internetwork address of the second host computer;
- (2) in the first bridge computer, intercepting the first data packet and determining whether the first and second host computers are among the predetermined plurality of host computers for which security is required, and if not, proceeding to step 5, and if so, proceeding to step 3;
- (3) encrypting the first data packet in the first bridge computer;
- (4) in the first bridge computer, generating and appending to the encrypted first data packet an encapsulation header, including:
  - (a) key management information [identifying] providing a mechanism for identifying the predetermined encryption method, and
  - (b) a new address header representing the source and destination for the first data packet, hereby generating a modified first data packet;
- (5) transmitting the first data packet or the modified first data packet from the first bridge computer via the internetwork to the second computer network;
- (6) intercepting the first data packet or the modified first data packet at the second bridge computer;
- (7) in the second bridge computer, if the encapsulation header has been appended to the first data packet, reading the encapsulation header, and determining

therefrom whether the first data packet was encrypted, [and if not, proceeding to step 10, and if so, proceeding to step 8] and if it is determined that the first data packet has been encrypted, proceeding to step 8 and otherwise proceeding to step 10;

- (8) in the second bridge computer, determining which encryption mechanism was used to encrypt the first data packet;
- (9) decrypting the first data packet by the second bridge computer;
- (10) transmitting the first data packet from the second bridge computer to the second host computer[,]; and
- (11) receiving the unencrypted first data packet at the second host computer.

2. (Once Amended) The method of claim 1, wherein the new address header for the modified first data packet includes the address of the second bridge computer.

3. (Once Amended) The method of claim 2, wherein the new address header for the modified first data packet includes an identifier of the second bridge computer.

4. (Once Amended) The method of claim 1, wherein the new address header of the modified first data packet includes the address of the second host computer.

5. (Once Amended) The method of claim 4, wherein the new address header for the modified first data packet includes an identifier of the second bridge computer.

6. (Once Amended) A system for automatically encrypting and decrypting data packets transmitted from a first host computer on a first computer network to a second host computer on a second computer network, including:

a first bridge computer coupled to the first computer network for intercepting data packets transmitted from said first computer network, the first bridge computer including a first processor and a first memory storing instructions for executing encryption of data packets according to a predetermined encryption/decryption mechanism;

a second bridge computer coupled to the second computer network for intercepting data packets transmitted to said second computer network, the second

bridge computer including a second processor and a second memory storing instructions for executing decryption of the data packets;

said first host computer including a third processor and a third memory including instructions for transmitting a first [said] data packet from said first host to said second host;

a first table stored in said first memory including a correlation of at least one of the first host computer and the first network with one of the second host computer and the second network, respectively;

instructions stored in said first memory for intercepting said first data packet before departure from said first network, determining whether said correlation is present in said first table, and if so, then executing encryption of said first data packet according to said predetermined encryption/decryption mechanism, generating a new address header including a mechanism for identifying said predetermined encryption/decryption mechanism and appending said new address header to said encrypted first data packet, thereby generating a modified first data packet, and transmitting said modified first data packet on to the second host computer;

a second table stored in said second memory including a correlation of at least one of the first host computer and the first network with one of the second host computer and the second network, respectively; and

instructions stored in said second memory for intercepting said modified first data packet upon arrival at said second network, determining whether said correlation is present in said second table, and if so, then executing decryption of said first data packet according to said predetermined encryption/decryption mechanism, and transmitting the first data packet to the second host computer.

7. (Once Amended) [The method of claim 6,] A system for automatically encrypting and decrypting data packets transmitted from a first host computer on a first computer network to a second host computer on a second computer network, including:

a first bridge computer coupled to the first computer network for intercepting data packets transmitted from said first computer network, the first bridge computer including a first processor and a first memory storing instructions for executing encryption of data packets according to a predetermined encryption/decryption mechanism;

\_\_\_\_\_ a second bridge computer coupled to the second computer network for intercepting data packets transmitted to said second computer network, the second bridge computer including a second processor and a second memory storing instructions for executing decryption of the data packets;

\_\_\_\_\_ said first host computer including a third processor and a third memory including instructions for transmitting a first data packet from said first host to said second host;

\_\_\_\_\_ a first table stored in said first memory including a correlation of at least one of the first host computer and the first network with one of the second host computer and the second network, respectively;

\_\_\_\_\_ instructions stored in said first memory for intercepting said first data packet before departure from said first network, determining whether said correlation is present in said first table, and if so, then executing encryption of said first data packet according to said predetermined encryption/decryption mechanism, generating a new address header and appending said new address header to said encrypted first data packet, thereby generating a modified first data packet, and transmitting said modified first data packet on to the second host computer, wherein said new address header includes [the] internetwork broadcast addresses of the first and second computer networks[.];

\_\_\_\_\_ a second table stored in said second memory including a correlation of at least one of the first host computer and the first network with one of the second host computer and the second network, respectively; and

\_\_\_\_\_ instructions stored in said second memory for intercepting said modified first data packet upon arrival at said second network, determining whether said correlation is present in said second table, and if so, then executing decryption of said first data packet according to said predetermined encryption/decryption mechanism, and transmitting the first data packet to the second host computer.

8. The method of claim 7, wherein said new address header includes an identifier of the second bridge computer.

9. The method of claim 6, wherein said new address header includes the address

of the second host computer.

10. The method of claim 9, wherein said new address header includes an identifier of the second bridge computer.

11. (Once Amended) A method for transmitting and receiving packets of data via an internetwork from a first host computer on a first computer network to a second host computer on a second computer network, [the first and second computer networks,] each of said first and second host computer networks, each of said first and second host computers including a processor and a memory for storing instructions for execution by the processor, each said memory storing at least [on] a predetermined encryption/decryption mechanism and a source/destination table identifying a predetermined plurality of sources and destinations requiring security for packets transmitted between them, the method being carried [carded] out by means of the instructions stored in said respective memories and including the steps of:

- (1) generating, by the first host computer, a first data packet for transmission to the second host computer, a portion of the first data packet including information representing an internetwork address of a source of the first data packet and an internetwork address of a destination of the first data packet;
- (2) in the first host computer, determining whether the source and destination of the first data packet are among the predetermined plurality of sources and destinations identified in said source/destination table for which security is required, and if not, proceeding to step 5, and if so, proceeding to step 3;
- (3) encrypting the first data packet in the first host computer;
- (4) in the first host computer, generating and appending to the encrypted first data packet an encapsulation header, including:
  - (a) key management information providing a mechanism for identifying the predetermined encryption method, and
  - (b) a new address header identifying the source and destination for the first data packet, hereby generating a modified first data packet;
- (5) transmitting the first data packet or the modified first data packet from the first host computer via the internetwork to the second computer network;
- (6) in the second host computer, if the encapsulation header has been appended to

the first data packet, reading the encapsulation header, and determining therefrom whether the first data packet was encrypted, and if the first data packet was not encrypted [not], ending the method, and if [so]the first data packet was encrypted, proceeding to step 7;

- (7) in the second host computer, determining which encryption mechanism was used to encrypt the first data packet; and
- (8) decrypting the first data packet by the second host computer.

12. (Once Amended) The method of claim 11, wherein the new address header for the modified first data packet includes internetwork broadcast addresses of the first and second computer networks.

13. The method of claim 11, wherein the source/destination table includes data identifying internetwork addresses of the first and second host computers.

14. (Once Amended) A system for automatically encrypting and decrypting data packets transmitted from a first host computer on a first computer network [and having a first host computer on a first computer network and] , the first host computer having a first processor and a first memory, via an internetwork to a second host computer on a second computer network [and having a second host computer on a second computer network and] , the second host computer having a second processor and a second memory, the system including:

security data stored in said first and second memories indicating that data packets meeting at least one predetermined criterion are to be encrypted;

a predetermined encryption/decryption mechanism stored in said first and second memories;

a decryption key stored in said second memory;

instructions stored in said first memory for determining whether to encrypt one or more data packets, by determining whether said at least one predetermined criterion is met by said one or more data packets [data packet];

instructions stored in said first memory for executing encryption according to said predetermined encryption/decryption mechanism of at least a first [said data packet] one of said one or more data packets, when said at least one predetermined

criterion is met, for generating a new address header for said first data packet and for appending an encapsulation header to said first data packet and transmitting said first data packet to said second host, said new address header identifying broadcast addresses of the first and second computer networks, said encapsulation header including at least said new address header; and

instructions stored in said second memory for receiving said first data packet, determining whether it has been encrypted by reference to said security data in said second memory, and if so then determining which encryption/decryption mechanism was used for encryption, and decrypting said first data packet by use of said decryption key.

15. (Once Amended) The system of claim 14, wherein:

said security data comprises correlation data stored in each of said first and second memories [identifying at least one of said first and second memories] identifying at least one of said first host computer and said first network correlated with at least one of said second host computer and said second network;

the system further including instructions stored in said first memory for determining whether to encrypt data packets by inspecting for a match between source and destination addresses of said data packets with said correlation data.

16. (Once Amended) A system for automatically encrypting data packets for transmission from a first host computer on a first computer network to a second host computer on a second computer network, said first host computer including a first processor and a first memory including instructions for transmitting said data packets from said first host to said second host, the system including:

a bridge computer coupled to the first computer network for intercepting at least a first [said] data packet transmitted from said first computer network, said bridge computer including a second processor and a second memory storing instructions for executing encryption of said first data packet according to a predetermined encryption/decryption mechanism;

information stored in said second memory correlating at least one of the first host computer and the first network with one of the second host computer and the second network, respectively; and



instructions stored in said second memory for intercepting said first data packet before departure from said first network, determining whether said correlation is present, and if so, then executing encryption of said first data packet according to said predetermined encryption/decryption mechanism, generating a new address header including a mechanism for identifying said predetermined encryption/decryption mechanism and appending said new address header to said first data packet, thereby generating a modified first data packet on to the second host computer.

17. (Once Amended) A method for transmitting packets of data via an internetwork from a first host computer on a first computer network to a second host computer on a second computer network, the first computer networks including a first bridge computer, each of said first and second host computers and said bridge computer further including memory storing at least one predetermined encryption/decryption mechanism and information identifying a predetermined plurality of host computers as hosts requiring security for packets transmitted between them, the method being carried out according to the instructions stored in said respective memories and including the steps of:

- (1) generating, by the first host computer, a first data packet for transmission to the second host computer, a portion of the first data packet including information representing an internetwork address of the first host computer and an internetwork address of the second host computer.
- (2) in the first bridge computer, intercepting the first data packet and determining whether the first and second host computers are among the predetermined plurality of host computers for which security is required, and if not, proceeding to step 5, and if so, proceeding to step 3;
- (3) encrypting the first data packet in the first bridge computer;
- (4) in the first bridge computer, generating and appending to the first data packet an encapsulation header, including:
  - (a) key management information providing a mechanism for identifying the predetermined encryption method, and
  - (b) a new address header representing the source and destination for the data packet, thereby generating a modified first data packet; and
- (5) transmitting the first data packet or the modified first data packet from the first

bridge computer via the internetwork to the second computer network.

18. (Once Amended) A system for automatically decrypting data packets transmitted from a first computer to a second computer, the system comprising:

a bridge coupled to the second computer for intercepting a data packet from the first computer, the data packet having an address header and a body, the address header including broadcast addresses of the first and second computers, the bridge including a processor and a memory that stores instructions for decrypting data packets;

information stored in the memory of the bridge correlating the first and second computers; and

instructions stored in the memory for intercepting the data packet, determining whether the information stored in the memory of the bridge correlates the first and second computers, and if so, decrypting at least a portion of the data packet to generate a new data packet including a new address header, and transmitting the new data packet onto the second computer.

19. (Once Amended) The system of claim 18, wherein the data packet includes the new data packet in encrypted form.

20. (Twice Amended) A system for automatically decrypting data packets transmitted from a first computer to a second computer, the system comprising:

a bridge coupled to the second computer for intercepting a data packet from the first computer, the data packet including a header storing key management information providing a mechanism for identifying an encryption method used to encrypt the data packet, the bridge including a processor and a memory that stores instructions for decrypting data packets;

information stored in the memory of the bridge correlating the first and second computers; and

instructions stored in the memory for intercepting the data packet, determining whether the information stored in the memory of the bridge correlates the first and second computers, and if so, decrypting the data packet to generate a new data packet including a new address header, and transmitting the new data packet onto the second

computer.

\_\_\_\_ 21.     The method of claim 18, wherein the new address header includes information indicating the first computer is a source of the new data packet and the second computer is a destination of the new data packet.

\_\_\_\_ 22.     (Once Amended)     A method for receiving data packets from a first computer to a second computer through a bridge including a processor and a memory that stores instructions for decrypting data packets and information correlating the first and second computers, the method being carried out according to instructions in the memory of the bridge and comprising:

\_\_\_\_ intercepting a data packet from the first computer to the second computer, the data packet including an address header and a body, the address header including broadcast addresses of the first and second computers and the body including address information representing an internetwork address of the first computer and an internetwork address of the second computer, wherein the address information is encrypted;

\_\_\_\_ determining whether the information stored in the memory of the bridge correlates the first and second computers, and if so, decrypting the data packet to generate a new data packet including a new address header; and

\_\_\_\_ transmitting the new data packet on to the second computer.

\_\_\_\_ 23.     (Once Amended)     The method of claim 22, wherein the body includes the new data packet in encrypted form.

\_\_\_\_ 24.     (Once Amended)     A method for receiving data packets from a first computer to a second computer through a bridge including a processor and a memory that stores instructions for decrypting data packets and information correlating the first and second computers, the method being carried out according to instructions in the memory of the bridge and comprising:

\_\_\_\_ intercepting a data packet from the first computer to the second computer, the data packet including information representing an internetwork address of the first computer and an internetwork address of the second computer;

determining whether the information stored in the memory of the bridge correlates the first and second computers, and if so, decrypting the data packet to generate a new data packet including a new address header; and  
transmitting the new data packet on to the second computer;  
wherein the data packet includes a header storing key management information providing a mechanism for identifying an encryption method used to encrypt the new data packet.

25. The method of claim 22, wherein the new address header includes information indicating the first computer is a source of the new data packet and the second computer is a destination of the new data packet.

26. (Once Amended) A method of encrypting data packets, comprising:  
receiving a data packet from a source for a destination, the data packet including a header section and a data section, the header section storing a source identifier and a destination identifier;

determining whether the data packet should be encrypted upon reference to at least one of the source and destination identifiers;

if the data packet should be encrypted, encrypting the data packet to produce an encrypted data packet; and

generating a new address header and appending the new address header to the encrypted data packet, thereby generating a modified data packet;

wherein the new address header includes a mechanism for identifying an encryption method used to generate the encrypted data packet.

27. (Once Amended) The method of claim 26, further comprising transmitting the modified data packet to the destination.

28. The method of claim 26, wherein the determining whether the data packet should be encrypted comprises accessing stored information that indicates by presence or absence of the source identifier that data packets from the source should be encrypted.

29. The method of claim 26, wherein the determining whether the data packet

should be encrypted comprises accessing stored information that indicates by presence or absence of a correlation between the source and destination identifiers that data packets from the source for the destination should be encrypted.

\_\_\_\_\_ 30. (Once Amended) The method of claim 26, wherein the encrypted data packet includes an encrypted data packet header section and an encrypted data packet data section, the encrypted data packet header section including the header section of the data packet after encryption and the encrypted data packet data section including the data section of the data packet after encryption, the modified data packet including a header portion storing the new address header and a data portion storing the encrypted data packet.

\_\_\_\_\_ 31. The method of claim 30, wherein the encrypted data packet header section stores the source and destination identifiers.

32. (Once Amended) A method of encrypting data packets, comprising:  
\_\_\_\_\_ receiving a data packet from a source for a destination, the data packet including a header section and a data section, the header section storing a source identifier and a destination identifier;

\_\_\_\_\_ determining whether the data packet should be encrypted upon reference to at least one of the source and destination identifiers;

\_\_\_\_\_ if the data packet should be encrypted, encrypting the data packet to produce an encrypted data packet; and

\_\_\_\_\_ generating a new address header and appending the new address header to the encrypted data packet, thereby generating a modified data packet;

\_\_\_\_\_ wherein the encrypted data packet includes an encrypted data packet header section and an encrypted data packet data section, the encrypted data packet header section including the header section of the data packet after encryption and the encrypted data packet data section including the data section of the data packet after encryption, the modified data packet including a header portion storing the new address header and a data portion storing the encrypted data packet;

\_\_\_\_\_ wherein the source is a host computer in a network and the header portion of the modified data packet stores an identifier of the network.

33. (Once Amended) A method of encrypting data packets, comprising:  
receiving a data packet from a source for a destination, the data packet including a  
header section and a data section, the header section storing a source identifier and a  
destination identifier;  
determining whether the data packet should be encrypted upon reference to at least  
one of the source and destination identifiers;  
if the data packet should be encrypted, encrypting the data packet to produce an  
encrypted data packet; and  
generating a new address header and appending the new address header to the  
encrypted data packet, thereby generating a modified data packet;  
wherein the encrypted data packet includes an encrypted data packet header section  
and an encrypted data packet data section, the encrypted data packet header section including  
the header section of the data packet after encryption and the encrypted data packet data  
section including the data section of the data packet after encryption, the modified data  
packet including a header portion storing the new address header and a data portion storing  
the encrypted data packet;  
wherein the destination is a host computer in a network and the header portion of the  
modified data packet stores an identifier of the network.

34. The method of claim 26, wherein the source is a host computer or a network.

35. The method of claim 26, wherein the destination is a host computer or a  
network.

36. (Once Amended) A computer program product adapted for encrypting  
data packets, comprising:  
computer code that when executed causes the reception of a data packet from a source  
for a destination, the data packet including a header section and a data section, and the header  
section storing a source identifier and a destination identifier;  
computer code that when executed causes the determination of whether the data  
packet should be encrypted upon reference to at least one of the source and destination  
identifiers;  
computer code that when executed, if the data packet should be encrypted, causes the

encryption of the data packet to produce an encrypted data packet;

computer code that when executed causes the generation of a new address header and appends the new address header to the encrypted data packet, the new address header including a mechanism for identifying an encryption method used to generate the encrypted data packet, thereby generating a modified data packet; and

a computer readable medium that stores the computer codes.

37. The computer program product of claim 36, wherein the computer readable medium is a memory, random-access-memory, read-only-memory, disk drive, or CD-ROM.

38. (Once Amended) A computer system for encrypting data packets, comprising:

a processor;

a computer readable medium coupled to the processor and storing a computer program comprising:

computer code that when executed by the processor causes the processor to receive a data packet from a source for a destination, the data packet including a header section and a data section, and the header section storing a source identifier and a destination identifier;

computer code that when executed by the processor causes the processor to determine whether the data packet should be encrypted upon reference to at least one of the source and destination identifiers;

computer code that when executed by the processor causes the processor to encrypt the data packet to produce an encrypted data packet when it is determined that the data packet should be encrypted; and

computer code that when executed by the processor causes the processor to generate a new address header and append the new address header to the encrypted data packet, thereby generating a modified data packet;

wherein the new address header includes a mechanism for identifying an encryption method used to generate the encrypted data packet.

39. The computer program product of claim 38, wherein the computer readable

medium is a memory, random-access-memory, read-only-memory, disk drive, or CD-ROM.

40. (Once Amended) A method of decrypting data packets, comprising:  
receiving a data packet from a source for a destination, the data packet including a  
header section and a data section, the header section storing a source identifier identifying a  
broadcast address of the source and a destination identifier identifying a broadcast address of  
the destination;  
determining whether the data packet is encrypted upon reference to at least one of the  
source and destination identifiers; and  
if the data packet is encrypted, decrypting the data packet to produce a decrypted data  
packet.

41. The method of claim 40, further comprising transmitting the decrypted data  
packet to the destination.

42. The method of claim 40, wherein the determining whether the data packet is  
encrypted comprises accessing stored information that indicates by presence or absence of  
the source identifier that data packets from the source are encrypted.

43. The method of claim 40, wherein the determining whether the data packet is  
encrypted comprises accessing stored information that indicates by presence or absence of a  
correlation between the source and destination identifiers that data packets from the source  
for the destination are encrypted.

44. (Once Amended) The method of claim 40, wherein the data section of the  
data packet includes an encrypted header section and an encrypted data section, the encrypted  
header section including a header of the decrypted data packet after encryption and the  
encrypted data section including a body of the decrypted data packet after encryption..

45. The method of claim 44, wherein the encrypted header section stores the  
source and destination identifiers.

46. The method of claim 44, wherein the source is a network and the encrypted



header section stores an identifier of a host computer in the network.

47. The method of claim 44, wherein the destination is a network and the encrypted header section stores an identifier of a host computer in the network.

48. The method of claim 40, wherein the source is a host computer or a network.

49. The method of claim 40, wherein the destination is a host computer or a network.

50. (Once Amended) A computer program product adapted for decrypting data packets, comprising:

computer code that when executed causes the reception of a data packet from a source for a destination, the data packet including a header section and a data section, and the header section storing a source identifier identifying a broadcast address of the source and a destination identifier identifying a broadcast address of the destination;

computer code that when executed causes the determination of whether the data packet is encrypted upon reference to at least one of the source and destination identifiers;

computer code that when executed and if the data packet is encrypted, causes the decryption of the data packet to produce a decrypted data packet; and

a computer readable medium that stores the computer codes.

51. The computer program product of claim 50, wherein the computer readable medium is a memory, random-access-memory, read-only-memory, disk drive, or CD-ROM.

52. (Once Amended) A computer system for decrypting data packets, comprising:

a processor;

a computer readable medium coupled to the processor and storing a computer program comprising:

computer code that when executed on the processor causes the processor to receive a data packet from a source for a destination, the data packet including a header section and a data section, the header section storing a source identifier

identifying a broadcast address of the source and a destination identifier identifying a broadcast address of the destination;

\_\_\_\_\_ computer code that when executed on the processor causes the processor to determine whether the data packet is encrypted upon reference to at least one of the source and destination identifiers; and

\_\_\_\_\_ computer code that when executed on the processor causes the processor to if the data packet is encrypted, decrypt the data packet to produce a decrypted data packet.

53. The computer program product of claim 52, wherein the computer readable medium is a memory, random access memory, read only memory, disk drive, or CD ROM.

54. A system for automatically encrypting and decrypting data packets transmitted from a first host computer on a first computer network, the first host computer having a first processor and a first memory, via an internetwork to a second host computer on a second computer network, the second host computer having a second processor and a second memory, the system including:

\_\_\_\_\_ security data stored in said first and second memories indicating that data packets meeting at least one predetermined criterion are to be encrypted;

\_\_\_\_\_ instructions stored in said first memory for determining whether to encrypt one or more data packets, by determining whether said at least one predetermined criterion is met by said one or more data packets;

\_\_\_\_\_ instructions stored in said first memory for executing encryption of at least a first one of said one or more data packets according to a predetermined encryption/decryption mechanism, when said at least one predetermined criterion is met, for generating a new address header for said first data packet and for appending an encapsulation header to said first data packet and transmitting said first data packet to said second host, said encapsulation header including said new address header and a mechanism for identifying said predetermined encryption/decryption mechanism;

\_\_\_\_\_ instructions stored in said second memory for receiving said first data packet,

determining whether it has been encrypted by reference to said security data in said second memory, and if so then determining which encryption/decryption mechanism was used for encryption, and decrypting said first data packet by use of said encryption/decryption mechanism.

55. The system as recited in claim 54, wherein said predetermined encryption/decryption mechanism is provided in encrypted form within said encapsulation header.

56. The system of claim 15, wherein said correlation data includes:  
encryption rules identifying source and destination networks to and from which  
packets are to be encrypted; and  
host information indicating exceptions to the encryption rules.

57. A system for automatically encrypting data packets for transmission from a first host computer on a first computer network to a second host computer on a second computer network, said first host computer including a first processor and a first memory including instructions for transmitting said data packets from said first host to said second host, the system including:

a bridge computer coupled to the first computer network for intercepting at least a first data packet transmitted from said first computer network, said bridge computer including a second processor and a second memory storing instructions for executing encryption of said first data packet according to a predetermined encryption/decryption mechanism;

information stored in said second memory correlating at least one of the first host computer and the first network with one of the second host computer and the second network, respectively; and

instructions stored in said second memory for intercepting said first data packet before departure from said first network, determining whether said correlation is present, and if so, then executing encryption of said first data packet according to said predetermined encryption/decryption mechanism, generating a new address header including the internetwork broadcast addresses of the first and second computer networks and appending said new address header to said first data packet, thereby generating a modified first data packet on to the second host computer.

58. A computer program product adapted for encrypting data packets, comprising:

computer code that when executed on a computer causes the computer to receive a data packet from a source for a destination, the data packet including a header section and a data section, and the header section storing a source identifier and a destination identifier;

computer code that when executed on a computer causes the computer to determine whether the data packet should be encrypted upon reference to at least one of the source and destination identifiers;

computer code that when executed on a computer causes the computer to, if the data packet should be encrypted, encrypt the data packet to produce an encrypted data packet;

computer code that when executed on a computer causes the computer to generate a new address header storing at least one of a broadcast address associated with the source and a broadcast address associated with the destination, and append the new address header to the encrypted data packet, thereby generating a modified data packet; and

a computer readable medium that stores the computer codes.

59. A computer system for encrypting data packets, comprising:

a processor;

a computer readable medium coupled to the processor storing a computer program comprising:

computer code that when executed by the processor causes the processor to receive a data packet from a source for a destination, the data packet including a header section and a data section, the header section storing a source identifier and a destination identifier;

computer code that when executed by the processor causes the processor to determine whether the data packet should be encrypted upon reference to at least one of the source and destination identifiers;

computer code that when executed by the processor causes the processor to if the data packet should be encrypted, encrypt the data packet to produce an encrypted data packet; and

computer code that when executed by the processor causes the processor to generate a new address header storing at least one of a broadcast address associated

the source and a broadcast address associated with the destination, and append the new address header to the encrypted data packet, thereby generating a modified data packet.

60. (Once Amended) A method of decrypting data packets, comprising:  
receiving a data packet from a source at a destination, the data packet including a header section and a data section, the header section storing a source identifier, a destination identifier, and encryption information providing a mechanism for identifying an encryption method used to generate the data packet; and  
decrypting the data packet to produce a decrypted data packet.

61. The method as recited in claim 60, further comprising:  
determining from the header section whether the data packet is encrypted; and  
wherein decrypting the data packet to produce a decrypted data packet is performed if it is determined that the data packet is encrypted.

62. The method as recited in claim 60, wherein decrypting the data packet to produce a decrypted data packet comprises:  
decrypting at least one of the data section of the data packet and the encryption information.

63. The method as recited in claim 60, wherein the data section includes a packet header and a packet body, and wherein decrypting the data section of the data packet comprises decrypting at least one of the packet header and the packet body.

64. (Once Amended) A computer program product adapted for decrypting data packets, comprising:  
computer code that when executed on a computer causes the computer to receive a data packet from a source at a destination, the data packet including a header section and a data section, the header section storing a source identifier, a destination identifier and encryption information including a mechanism for identifying an encryption method used to

generate the data packet;

computer code that when executed on a computer causes the computer to decrypt the data packet to produce a decrypted data packet; and

a computer readable medium that stores the computer codes.

65. The computer program product as recited in claim 64, further comprising:

computer code that when executed on a computer causes the computer to determine from the header section whether the data packet is encrypted; and

computer code that when executed on a computer causes the computer to decrypt the data packet if it is determined that the data packet is encrypted.

66. The computer program product as recited in claim 64, further comprising:

computer code that when executed on a computer causes the computer to decrypt the data packet using the encryption method.

67. (Once Amended) A computer system for decrypting data packets, comprising:

a processor;

a computer readable medium coupled to the processor storing a computer program comprising:

computer code that when executed on the processor causes the processor to receive a data packet from a source at a destination, the data packet including a header section and a data section, the header section storing a source identifier, a destination identifier and encryption information including a mechanism for identifying an encryption method used to generate the data packet;

computer code that when executed on the processor causes the processor to determine from the header section whether the data packet is encrypted; and

computer code that when executed on the processor causes the processor to if the data packet is encrypted, decrypt the data packet to produce a decrypted data packet.

68. The computer system as recited in claim 67, further comprising:

computer code that when executed on a computer causes the computer to decrypt the data packet using the encryption method.

69. The system as recited in claim 16, wherein the mechanism indirectly references said predetermined encryption/decryption mechanism.

70. The system as recited in claim 20, wherein the mechanism indirectly identifies the encryption method.

71. The method as recited in claim 26, wherein the mechanism indirectly identifies the encryption method.

72. The computer program product as recited in claim 36, wherein the mechanism indirectly identifies the encryption method.

73. The computer system as recited in claim 38, wherein the mechanism indirectly identifies the encryption method.